Avaliação 2 - Regressão

Leonardo Ribeiro Damiani Júnior Cartão UFRGS: 00326165

Importação e Análise do Banco

Importação

O banco de estudantes foi importado e este possui 649 linhas por 33 colunas (variáveis) denominado `dados`.

Abaixo temos as primeiras linhas dos dados de estudantes de um certo local e algumas de suas colunas escolhidas junto da variável G3 (nossa variável alvo sendo a nota final dos alunos).

Tabela 1: Primeiras Seis Linhas e Algumas Colunas

school	sex	age	studytime	activities	internet	romantic	G3
GP	F	18	2	no	no	no	11
GP	F	17	2	no	yes	no	11
GP	F	15	2	no	yes	no	12
GP	F	15	3	yes	yes	yes	14
GP	F	16	2	no	no	no	13
GP	M	16	2	yes	yes	no	13

Análise do Banco

Podemos então, perceber que não existem dados faltantes em nosso banco:

Número de Dados Faltantes (NA)
0

Ainda, conseguimos verificar quantas variáveis qualitativas e quantitativas nós temos em nosso banco. Note que primeiro estamos transformando as devidas variáveis em fatores (categóricas) e já retirando as variáveis G1 e G2 do nosso banco (variáveis que compoem a nota final).

	Qualitativas	Qualitativas
Frequências	4	27

Assim, percebemos que nosso banco é formado em sua maioria por variáveis categóricas e por isso, iremos utilizar, posteriormente, o one-hot-encoding de forma a tratarmos estas categorias em pról dos futuros métodos que iremos ajustar como Random Forest, Bagging e Boosting.

Além disso, em nossa próxima página, ainda, estaremos apresentando as frequências destas nossas variáveis categóricas junto das principais estatísticas das nossas variáveis quantitativas para nos termos uma noção de como estas se comportam.

Tabela 4: Tabelas de Frequências das Variáveis Qualitativas

school	sex	address	famsize	Pstatus	schoolsup
GP:423	F:383	R:197	GT3:457	A: 80	no :581
MS:226	M:266	U:452	LE3:192	T:569	yes: 68

famsup	paid	activities	nursery	higher	internet	romantic
no :251	no :610	no :334	no :128	no : 69	no :151	no :410
yes:398	yes: 39	yes:315	yes:521	yes:580	yes:498	yes:239

Medu	Fedu	Mjob	Fjob	reason	guardian
0: 6	0: 7	at_home :135	at_home : 42	course :285	father:153
1:143	1:174	health : 48	health : 23	home :149	mother:455
2:186	2:209	other :258	other :367	other : 72	other : 41
3:139	3:131	services:136	services:181	reputation:143	NA
4:175	4:128	teacher : 72	teacher : 36	NA	NA

traveltime	studytime	famrel	freetime	goout	Dalc	Walc	health
1:366	1:212	1: 22	1: 45	1: 48	1:451	1:247	1: 90
2:213	2:305	2: 29	2:107	2:145	2:121	2:150	2: 78
3: 54	3: 97	3:101	3:251	3:205	3: 43	3:120	3:124
4: 16	4: 35	4:317	4:178	4:141	4: 17	4: 87	4:108
NA	NA	5:180	5: 68	5:110	5: 17	5: 45	5:249

Tabela 8: Estatísticas Descritivas das Variáveis Numéricas

	Mínimo	Máximo	Mediana	Média	Variância	Desvio Padrão	Coeficiente de Variação
age	15	22	17	16.7442	1.4839	1.2181	0.0727
failures	0	3	0	0.2219	0.3519	0.5932	2.6737
absences	0	32	2	3.6595	21.5366	4.6408	1.2681
G3	0	19	12	11.9060	10.4371	3.2307	0.2713

Através das tabelas acima, podemos perceber que nas variáveis catégoricas algumas delas tem poucas observações, sendo estas as variáveis: Medu, Fedu, traveltime, famrel, Dalc. Por isso, iremos verificar as suas relações com a variável resposta G3,

Já para as variáveis numéricas, percebemos um comportamento curioso nas variáveis failures e absences, pois ambas parecem se concentrar em valores baixos como em zero. Então, também olharemos como estas variáveis se comportam em relação a G3.

Ressaltamos, que as demais variáveis seguirão como estão.

Análise de Medu, Fedu, traveltime, famrel, Dalc, failures e absences.

Abaixo, estaremos agrupando as variáveis categórias pelas suas respectivas categorias e olhando para a média destes índividuos (de cada categoria) na variável G3.

Categorias - Medu	G3
0	11.667
1	10.797
2	11.661
3	11.921
4	13.069

Categorias - Fedu	G3
0	12.143
1	10.937
2	11.785
3	12.382
4	12.922

Categorias - traveltime	G3
1	12.251
2	11.577
3	11.167
4	10.875

Categorias - famrel	G3
1	10.636
2	10.862
3	11.594
4	12.344
5	11.633

Categorias - Dalc	G3
1	12.299
2	11.364
3	11.140
4	8.941
5	10.235

Assim, através das tabelas percebemos que podemos realizar algumas modificações em nosso conjunto de dados.q (juntar categorias), pois note algumas das nossas categorias que apresentaram poucos valores possuem uma média parecida com pelo menos uma outra categoria.

Por isso, para reforçar essa nossa análise estaremos apresentando as modificações que iremos fazer nas variáveis, onde os números entre parênteses são as frequências de cada categorias junto das médias da variável G3.

Modificações nas Categóricas

- **Medu:** Estamos juntando as categorias 0 (6 | 11.667) e 2 (186 | 11.661) na categoria 0e2.
- **Fedu:** Estamos juntando as categorias 0 (7 | 12.143) e 3 (131 | 12.382) na categoria 0e3.
- **traveltime:** Estamos juntando as categorias 3 (54 | 11.167) e 4 (16 | 10.875) na categoria 3e4.
- **famrel:** Estamos juntando as categorias 1 (22 | 10.636) e 2 (29 | 10.862) na categoria 1e2.
- **Dalc:** Não iremos juntar nenhuma das categorias 3 (43 | 11.14), 4 (17 | 8.941) e 5 (17 | 10.235).

Aqui reforçamos que estas modificações são buscando aumentar as informações para os nossos métodos que virão em seguida e como podemos perceber acima, decidimos por não juntar as categorias da variável Dalc, pois estas apresentavam valores muito diferentes para a média.

Agora olharemos para as nossas variáveis numéricas. Primeiramente utilizaremos as tabela de frequências novamente para verificar o comportamento dessas.

Tabela 14: Frequências de Failures

0	1	2	3
549	70	16	14

Tabela 15: Frequências de Absences

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	18	21	22	24	26	30	32
244	12	110	7	93	12	49	3	42	7	21	5	12	1	8	2	10	3	2	2	1	1	1	1

Percebemos, em ambas tabelas, que parece haver uma concentração em valores baixos, como no caso da variável failures, a maioria de suas observações foram de valor zero. Assim, podemos transformar estas variáveis em categóricas no mesmo intuito que anteiormente, buscando enriquecer nossos métodos.

Modificações nas Variáveis Numéricas

Sendo assim, avaliamos como se comportam as médias dessas possíveis categorias nas nossas variáveis de interesse.

Categorias - failures	G3
0	12.510
1	8.643
2	8.812
3	8.071

Categorias - absences	G3
0	12.041
1	12.417
2	12.191
3	10.429
4	12.011
5	11.750
6	12.122
7	13.000
8	11.619
9	9.714
10	12.238
11	11.200
12	10.083
13	14.000
14	10.375
15	11.000
16	10.300
18	12.333
21	11.500
22	8.000
24	9.000
26	8.000
30	16.000
32	14.000

Como procedemos no caso das categóricas, transformaremos as variáveis em categóricas agregando alguma de suas categorias, mas note que procederemos isto apenas na variável failures. Por isso, temos as seguintes conclusões:

- **failures:** Estamos juntando as observações 1 (70 | 8.643), 2 (16 | 8.812) e 3 (14 | 8.071) na categoria 1e2e3.
- **absences:** Não juntaremos as observações em categorias, pois os valores em média para G3 são muito diferentes uns dos outros.

Portanto, ressaltamos que estas análises nestas variáveis, tanto nas categóricas quanto nas numéricas, estão considerando uma possível melhora dos nossos resultados (EQM) que será abordada mais a frente com as implementações.

Além disso, as decisões sobre não alterar as variáveis Dalc e absences é reforçada quando nós fizemos as alterações deixadas em comentários nos códigos anteriores, pois os resultados apresentaram um efeito negativo, ou seja, o desempenho de nossos métodos pioraram.

Assim, com estas análises em nosso banco de dados, passamos agora para a separação do banco, seguindo da aplicação do one-hot-encoding, antes de ajustarmos o nosso modelo linear.

Separação do Banco de Dados

Nesta parte iremos particionar o nosso banco de dados em 70 % para treino e 30 % para a predição como orientado no enunciado da nossa atividade. Através do código abaixo estamos realizando a partição a partir da semente do número do meu cartão UFRGS.

```
# Semente (Cartão UFRGS)
set.seed(326165)

# Índices para a separação
temp = sample(1:dim(dados)[1], size = 0.7*dim(dados)[1], replace = F)

# Conjunto de Treino
dados.t = dados[temp,]

# Conjunto de Predição
dados.p = dados[-temp,]
```

Assim podemos notar abaixo que as dimensões dos nossos conjuntos de treino e predição.

Tabela 18: Dimensões dos Dados

	Linhas	Colunas
Treino	454	31
Predição	195	31

Note que a variável `temp` possui os índices das observações aleatorizadas para cada banco.

One – Hot – Encoding

Nesta seção estamos interessados em utilizar o One-Hot-Encoding para obtermos os mesmos bancos de treino e predição com as devidas mudanças nas variáveis categórias (valores númericos 1 e 0).

O código abaixo está explicando o que está acontecendo em nosso banco.

```
### ONE-HOT-ENCODING (automatizado)

dados.hot = data.frame(rep(NA, 649)) # novo banco
n = length(dados)

for (i in 1:n){

  # Nome da coluna
  name = colnames(dados)[i]
  # Verifica se a coluna é um fator
  if (class(dados[, i]) != "factor"){next}

  # Faz o One - Hot - Enconding
  ohe <- model.matrix( ~ dados[, i] -1) # separa em outras variaveis dummys

  # Colocando as variaveis no banco para o modelo
  for (k in 1:(ncol(ohe) - 1)){
     # O nome da nova variável será o nome antigo mais a categoria
     # representativa do número 1
     name2 = paste0(name,"_",levels(dados[,i])[k])
     dados.hot[,name2] = ohe[,k]
  }}

# Tirando a primeira variável de NAs
dados.hot$rep.NA..649. = NULL
# Adcionando as variáveis quantitativas - adicionar as numéricas
dados.hot = cbind(dados.n[,c("age", "absences", "G3")], dados.hot)
```

Portanto, a partir do código acima, estamos com um novo banco denominado `dados.hot` no seguinte estilo (Novamente só mostramos as primeiras linhas).

Tabela 19: Banco dados.hot (Pós One - Hot - Encoding)

school_GP	sex_F	age	failures_0	romantic_no	absences	G3
1	1	18	1	1	4	11
1	1	17	1	1	2	11
1	1	15	1	1	6	12
1	1	15	1	0	0	14
1	1	16	1	1	0	13
1	0	16	1	1	6	13

Além disso, note ainda que estaremos utilizando a mesma separação que anteriormente para gerar os dados de treino (`dados.hot.t`) e predição (`dados.hot.p`).

Modelo Linear Explicativo com Stepwise

A seguir, através do R, estamos gerando um modelo linear no nosso conjunto de dados para treino sem one-hot-encoding e aplicando neste modelo a técnica de Stepwise para a escolha de um modelo mais parcimonioso.

```
## MODELO LINEAR PRIMEIRO
mod.lm = lm(G3 ~ ., data = dados.t)

## MODELO LINEAR COM STEPWISE
mod.lms = stepAIC(mod.lm, direction = 'both', trace = 0)
```

Portanto, percebemos com as funções acima que das nossas 30 variáveis explicativas, estamos utilizando apenas 14 em nosso modelo linear abaixo. Assim, calculamos os coeficientes do modelo.

```
## lm(formula = G3 ~ age + failures + school + sex + address + famsize +
##     Fedu + studytime + schoolsup + higher + famrel + goout +
##     Dalc + health, data = dados.t)
```

Tabela 20: Coeficientes do Modelo

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	6.6602966	2.1720283	3.0663950	0.0023052
age	0.2532411	0.1156774	2.1892016	0.0291267
failures1e2e3	-2.8267680	0.4063429	-6.9566075	0.0000000
schoolMS	-1.3049360	0.2953336	-4.4185145	0.0000126
sexM	-0.7506586	0.2828305	-2.6540938	0.0082512
addressU	0.5051114	0.2992925	1.6876847	0.0922072
famsizeLE3	0.4639325	0.2785777	1.6653613	0.0965793
Fedu1	-0.9301884	0.3752854	-2.4786163	0.0135774
Fedu2	-0.4855500	0.3558039	-1.3646564	0.1730849
Fedu4	0.3954258	0.4005423	0.9872259	0.3240949
studytime2	0.0661634	0.3090192	0.2141078	0.8305659
studytime3	0.7767188	0.4135986	1.8779532	0.0610737
studytime4	1.0819749	0.5875577	1.8414786	0.0662498
schoolsupyes	-1.5683481	0.4395397	-3.5681602	0.0004006
higheryes	1.7018375	0.4366097	3.8978459	0.0001128
famrel3	0.6763702	0.5768594	1.1725045	0.2416529
famrel4	1.3518823	0.5122223	2.6392491	0.0086152
famrel5	0.7608178	0.5383366	1.4132753	0.1583085
goout2	0.8423221	0.5777760	1.4578696	0.1456168
goout3	0.0404484	0.5562950	0.0727103	0.9420709
goout4	0.1331526	0.5751285	0.2315181	0.8170240
goout5	-0.3331081	0.5891264	-0.5654272	0.5720823
Dalc2	-0.6558458	0.3390363	-1.9344413	0.0537239
Dalc3	0.1399472	0.5415951	0.2583982	0.7962250
Dalc4	-2.4751605	0.7005282	-3.5332775	0.0004556
Dalc5	-0.3690416	0.8117656	-0.4546159	0.6496182
health2	0.2240408	0.5076729	0.4413093	0.6592140
health3	-0.6652248	0.4504254	-1.4768813	0.1404498
health4	-0.1435252	0.4401691	-0.3260683	0.7445335
health5	-0.9828821	0.3920783	-2.5068516	0.0125544

Interpretação dos Coeficientes Modelo

A partir do modelo gerado e seus coeficientes na página anterior, conseguimos interpretar como a nossa variável alvo G3 se comporta em relação as variáveis utilizadas no modelo, sendo esta uma das vantagens do modelo linear.

Além disso, ressaltamos que para os coeficientes estimados, nós poderíamos calcular os seus intervalos de confiança, de forma a termos assim uma medida de variabilidade desses coeficientes (não iremos apresentar os intervalos, pois não é o foco da atividade).

Sendo assim, para interpretarmos os nossos coeficientes, devemos ter em mente dois pontos chaves: o primeiro é que o nosso modelo calculado no R, leva em consideração, no caso de variáveis categóricas, a primeira categoria como de referência e além disso, as demais variáveis do banco de dados quando estivermos comparando categorias de uma certa variável, estão mantidas constantes.

Por isso, temos as seguintes relações para cada variável do modelo em comparação a G3.

- **age** – A variação esperada na nota do aluno, quando a idade deste aumenta em uma unidade é de 0.253. Assim, temos que quanto mais velho o aluno, este tende a ter uma nota maior quando consideramos a variável idade.
- **failures** – A nota do aluno, caso este já tenha reprovado de ano (nossa modificação de 1e2e3) é menor do que caso ele não tenha sido reprovado, pois estamos considerando a variável failures. Mantido as demais variáveis constantes, o aluno que já tenha reprovado tem uma nota estimada como a nota de alguém que passou - 2.827.
- **school** – A nota do aluno, caso este seja da escola Gabriel Pereira é maior do que caso ele seja da escola Mousinho da Silveira, pois estamos considerando a variável school. Mantido as demais variáveis constantes, o aluno que seja da escola Mousinho da Silveira tem uma nota estimada como a nota da escola Gabriel Pereira - 2.827.
- **sex** – A nota do aluno, caso este seja do sexo feminino é menor do que caso ele seja do sexo masculino, pois estamos considerando a variável sex. Mantido as demais variáveis constantes, o aluno que seja do sexo feminino tem uma nota estimada como a nota do sexo masculino - 1.305.
- **address** – A nota do aluno, caso este more na zona rural é maior do que caso ele more na zona urbana, pois estamos considerando a variável address. Mantido as demais variáveis constantes, o aluno que more na zona rural tem uma nota estimada como a nota do aluno da zona urbana + 0.505.
- **famsize** – A nota do aluno, caso este pertence a uma família de tamanho menor ou igual 3 é maior do que caso ele pertença a uma família de tamanho maior do que 3, pois estamos considerando a variável school. Mantido as demais variáveis constantes, o aluno que seja da família de tamanho menor ou igual 3 tem uma nota estimada como a nota da de um aluno da família de tamanho maior do que 3 + 0.464.
- **Fedu** – A nota do aluno, caso este tenha um pai com educação superior é maior do que caso ele tenha um pai com apenas ensino médio ou nenhuma educação, pois estamos considerando a variável school. Ainda, através da nossa variável de referência percebemos que caso o aluno tenha um pai com apenas ensino médio ou nenhuma educação, este possuirá uma nota maior do que os alunos com pai que estudaram do quinto até o nono ano ou mesmo os alunos que tem os pais com aenas o ensino fundamental (sendo estes alunos os com as piores notas).
- **studytime** – A nota do aluno, caso este dedique mais tempo de estudo durante a semana tende a aumentar, assim o aluno que tenha mais tempo dedicado de estudo durante a semana tem uma nota maior do que aquele aluno que dedicou menos tempo.
- **schoolsup** – A nota do aluno, caso este tenha um apoio educacional extra é menor do que caso ele não tenha um apoio educacional extra, pois estamos considerando a variável schoolsup. Mantido as demais variáveis constantes, o aluno que tenha apoio educacional extra tem uma nota estimada como a nota de quem não tem apoio educacional extra - 1.568.

- **higher** – A nota do aluno, caso este queira fazer o ensino superior é maior do que caso ele não queira fazer o ensino superior, pois estamos considerando a variável higher. Mantido as demais variáveis constantes, o aluno que queira fazer o ensino superior tem uma nota estimada como a nota de um aluno que não queira fazer o ensino superior + 1.702.
- **famrel** – A nota do aluno, caso este pertença uma família com relacionamentos ruins ou muito ruins é menor do que o aluno pertencesse a uma família com relacionamentos normais, bons ou muito bons. Ainda, pelos nossos coeficientes, percebemos que o aluno possuirá a maior nota caso este pertença a uma família com um bom relacionamento.
- **goout** – A nota do aluno, caso este saia com seus amigos numa frequência muito baixa ou normal ou até alta não parece variar muito, mas caso o aluno saia pouco com os seus amigos este tende a ter uma nota maior que as demais frequências e caso este saia numa frequência muito alta, este tende a ter a pior das notas.
- **Dalc** – A nota do aluno, caso este consuma muito pouco álcool diariamente ou numa quantidade normal não parece variar muito, mas caso o aluno consuma muito diariamente este tende a ter uma nota pior que os dois primeiros casos e caso este consuma pouco, este tende a ter pior nota ainda, só não sendo pior do que o aluno que consome de álcool elevada (não maior do que o muito).
- **health** – A nota do aluno tende a piorar na medida que o nível da sua saúde aumenta, com exceção do caso em que o aluno está mal, pois este tem uma nota maior do que quando ele está muito mal.

Erro Quadrado Médio do Modelo Linear

Por fim, utilizando o conjunto separado para a predição, realizamos a predição com nosso modelo linear afim de se calcular o Erro Quadrático Médio (EQM).

```
# Predição
pred.lms = predict(mod.lms, dados.p)

# Data frame de Comparação
comp = data_frame(predito = pred.lms, real = dados.p$G3)

# Funcao para servir no sumario dos treinos
eqm <- function(real, predito) {
  n = length(real)
  sum((real-predito)^2)/n
}

# Erro de Predição
eqm(comp$predito, comp$real)
```

```
## [1] 6.722438
```

Sendo assim, com este nosso modelo, chegamos a um $EQM = 6.722$.

Técnicas para a Regressão

Nesta seção estaremos ajustando diferentes técnicas na intenção de podermos preedizer a nossa variável G3 de forma a estarmos reduzindo o nosso EQM do modelo linear. Note que estamos utilizando todas as variáveis em nossas técnicas, pois estaremos buscando passar o máximo de informação à elas.

Ainda, sobre o banco `dados.hot` que será utilizado, temos que este possui como classes de referência, nas categóricas, as primeiras categorias se assemelhando da abordagem do modelo linear.

Random Forest

Assim, começamos primeiramente utilizando uma implementação do Random Forest através do pacote `randomForest`. Note que para um melhor desempenho do algoritmo estaremos utilizando o nosso banco `dados.hot.t`.

```
## RANDOM FOREST
mod.for = randomForest(G3 ~ ., data = dados.hot.t,
                       mtry = round((length(dados.hot.t) - 1)/3),  # regressao -> m = p/3
                       importance = TRUE,
                       ntrees = 1000)
pred.for = predict(mod.for, newdata = dados.hot.p)

# EQM random forest
eqm(pred.for, dados.hot.p$G3)
```

```
## [1] 6.37347
```

Sendo assim, com o Random Forest, chegamos a um $EQM = 6.373$ inferior ao encontrado pelo nosso modelo linear.

Assim, podemos resumir que esta queda em nosso erro quadrado médio seja devido a forma do procedimento do Random Forest que é baseada em Bootstrapp Aggregation (Bagging), o que o torna um algoritmo ensambled. Ou seja, a combinação de previsões de múltiplos algoritmos de machine learning juntos (diversas árvores), de forma a obter previsões mais acuradas do que qualquer modelo individual.

Portanto, na tentativa da minização do erro, deve-se ter em mente que a combinação de previsões de vários modelos em conjuntos funciona melhor se as previsões dos submodelos não forem correlacionadas. Por isso, o Random Forest altera o algoritmo para a maneira como as suas sub-árvores são aprendidas para que as previsões resultantes de todas as sub-árvores tenham menos correlação e consequentemente numa melhor acurácia (como dissemos).

Por isso, a utilização do bootstrap pelo algoritmo é para realizar uma reamostragem dos dados utilizados nestas árvores de decisão (sub-árvores). Assim, cada vez que uma divisão em uma árvore é considerada, uma amostra aleatória de m preditores são escolhidos como candidatos divididos do conjunto completo de preditores p, onde temos m $<$ p (nosso caso usamos m = p/3). Pois, assim o Random Forest força cada divisão de seus nós a considerar apenas um subconjunto dos preditores, o que acarretará na nossa procura pelas árvores menos correlacionadas.

Além disso, outra diferença importante para o modelo linear é a perda da nossa intepretação em nosso método, pois note que é inviável para a mente humana compreender as diversas árvores que estamos construindo através do Random Forest.

Uma vez que compreendemos um pouco melhor o procedimento do Random Forest, passamos ao Bagging.

Bagging

Agora modificaremos a nossa implementação do Random Forest para este, se transformar numa implementação do Bagging. Por isso, estaremos mudando os m, preditores escolhidos como candidatos divididos do conjunto completo de preditores p, para m = p.

Note que, novamente estamos utilizando o nosso banco `dados.hot.t`.

```
## BAGGING
mod.bag = randomForest(G3 ~ ., data = dados.hot.t,
                       mtry = (length(dados.hot.t) - 1),  # regression -> bagging -> m = p
                       importance = TRUE,
                       ntrees = 1000)
pred.bag = predict(mod.bag, newdata = dados.hot.p)

# EQM random forest
eqm(pred.bag, dados.hot.p$G3)
```

```
## [1] 6.979668
```

Sendo assim, com o Bagging, chegamos a um $EQM = 6.98$ superior ao encontrado pelo nosso modelo linear e pela Random Forest.

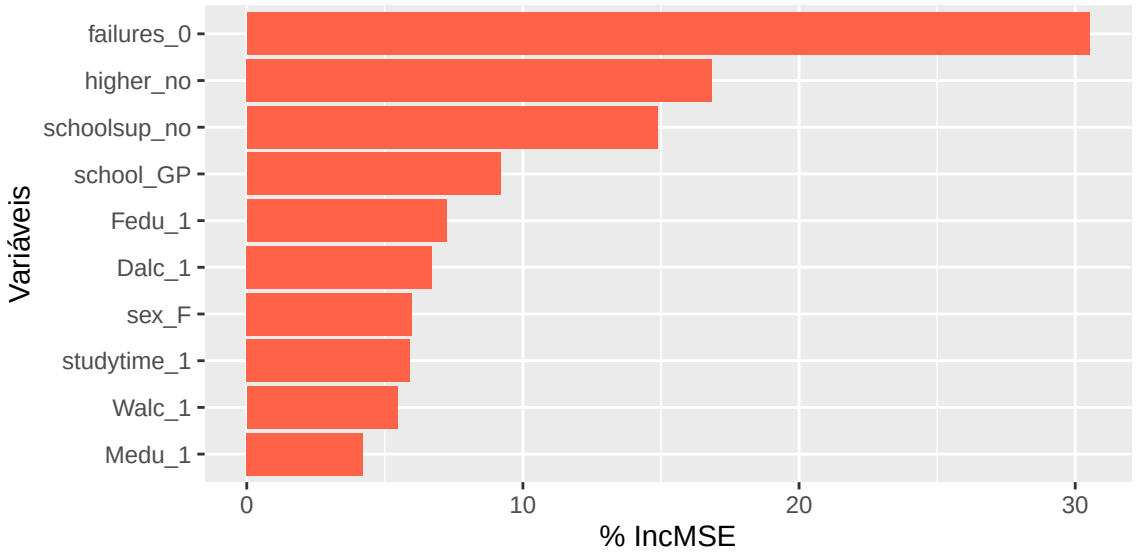
Como o Bagging neste caso trata do caso em que m = p, nós primeiro devemos lembrar que ele se comporta da mesma que a Random Forest com m < p, porém com essa diferença em relação a m, nós não temos a garantia da melhor (inferior) correlação das nossas árvores geradas em nossa floresta.

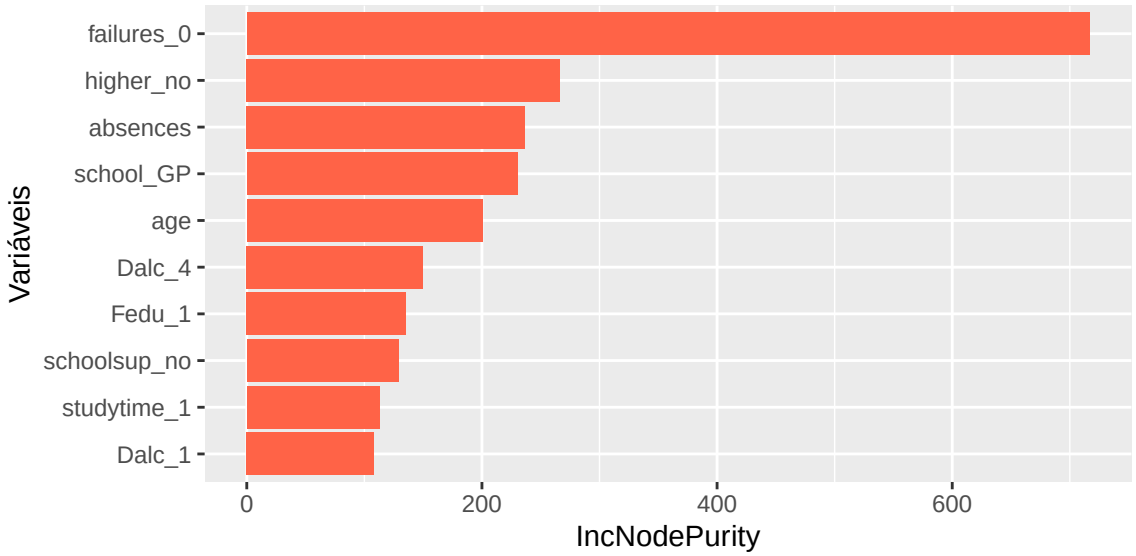
Sendo assim, tanto pela aleatorização do bootstrap quanto por essa diferença comentada, nós temos uma melhor performance neste caso quando comparada aos demais métodos já utilizados.

Comparação entre Random Forest e Bagging

Agora, como verificamos a melhor técnica dentre as duas utilizadas através do EQM, podemos notar abaixo as variáveis mais influentes neste nosso erro através da comparação de duas medidas.

- $\%IncMSE$ – baseia-se na diminuição média da precisão nas previsões sobre as amostras do out of bag quando uma dada variável é permutada.
- $IncNodePurity$ – uma medida da diminuição total na impureza do nó que resulta das divisões sobre essa variável, calculada sobre todas as árvores.





Com os gráficos acima, percebemos que temos as variáveis $Failures$ e $Heigher$ como as variáveis mais importantes dentre todas as árvores consideradas em nossa Random Forest.

Uma possível aplicação seria o ajustamento dos métodos utilizados aqui, utilizando estas variáveis mais importantes, considerando todas as principais influentes.

Boosting

Nesta última parte, estaremos buscando utilizar a técnica de Boosting. Com o código abaixo, buscamos mudar alguns parâmetros de forma a encontrarmos um melhor método para a predição de G3.

```
mod.boost <- gbm(G3 ~ ., data = dados.hot.t,
                 distribution = "gaussian",
                 n.trees = 5000,
                 interaction.depth = 2,
                 shrinkage = 0.001)
pred.boost = predict(mod.boost, newdata = dados.hot.p, n.trees = 5000)

# EQM random forest
eqm(pred.boost, dados.hot.p$G3)
```

```
## [1] 6.081407
```

Sendo assim, com o Boosting, chegamos a um $EQM = 6.08$ inferior aos demais encontrados, caracterizando assim como o Boosting como nossa melhor técnica agora.

Isso pode ser caracterizado pela forma como o algoritmo de Boosting opera. As árvores de nossa floresta são cultivadas sequencialmente, ou seja, cada árvore é cultivada usando informações de árvores já criadas. Boosting não envolve a reamostragem através do bootstrap, na verdade cada árvore é ajustada em uma versão modificada do conjunto de dados original.

Por isso, o Boosting ao contrário de ajustar uma única grande árvore de decisão aos dados, o que equivale a ajustar os dados de forma rígida e potencialmente overfitting, ele aprende lentamente, ou seja, se caracteriza por uma das abordagens de aprendizagem estatística que aprendem lentamente.

Assim, como fizemos com o Random Forest, nós podemos verificar as variáveis mais importantes no Boosting e plotarmos os seus gráficos de dependência parcial.

Complementação do Boosting

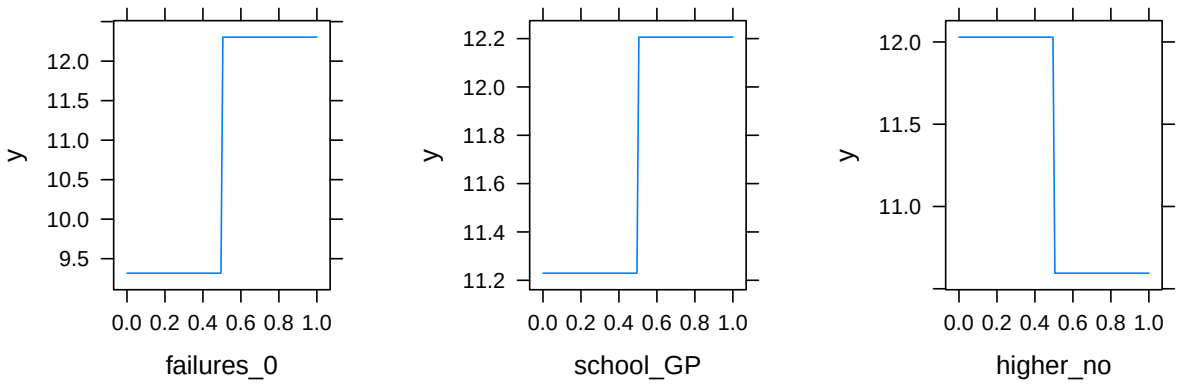
Primeiro, temos abaixo a tabela contendo as variáveis mais importantes.

Tabela 21: Variáveis Mais Importantes

var	rel.inf
failures_0	29.330876
school_GP	9.217672
higher_no	8.731789
Fedu_1	4.335246
absences	4.188927
Dalc_1	4.178060
schoolsup_no	3.700122
famrel_4	3.115809
sex_F	2.843911
studytime_1	2.663723
age	2.162412
Dalc_4	2.159401

Assim, percebemos que as variáveis mais importantes são as variáveis Failures, School e Higher. Note que, assim como aconteceu anteriormente, estas variáveis aparecem entre as mais importantes.

Por isso, plotaremos os gráficos para ilutrar os efeitos marginais dessas variáveis em relação a variável G3.



Portanto, podemos perceber que as notas aumentam com alunos não reprovados (failures_0), aumentam também com alunos vindo da escola Gabriel Pereira (school_GP) e as notas diminiuem com o não interesse em cursar o ensino superior.

Percebemos que estas conclusões se assemelham as encontradas para estas variáveis no modelo linear.

Conclusão

Ao final deste trabalho acredito ter compreendido o objetivo da diferença que podemos ter em relação aos erros de predição quando mudamos o foco da questão explicativa/interpretação para um foco na tentativa de minimizar os erros de predição.

Isto fica claro ao compararmos o Modelo Linear e as demais técnicas, onde não conseguimos nas demais uma interpretação clara como ocorre nos coeficientes do modelo. Além disso, a minização dos erros é o claro objetivo buscado pelas técnicas e por isso todas as análises tanto através do one-hot-encoding quanto nas junções de classes das categorias foram em pról de buscar este melhor resultado.

Reforço ainda que uma ideia inicial era utilizar a tunagem para tentar potencializar o nosso melhor método (Boosting), mas não foi possível a realização deste devido ao tempo para entregue desta atividade.